

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-179937

(43)Date of publication of application : 12.07.1996

(51)Int.Cl.

G06F 9/06
G06F 11/14

(21)Application number : 06-324423

(71)Applicant : CANON INC

(22)Date of filing : 27.12.1994

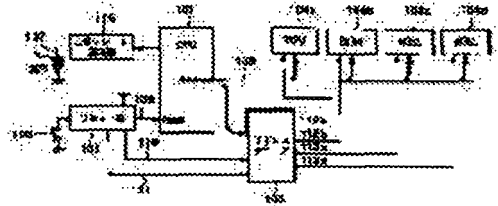
(72)Inventor : HONMA HIDEO

(54) DEVICE AND METHOD FOR SWITCHING BOOT PROGRAM

(57)Abstract:

PURPOSE: To provide a boot program switching device/method which can automatically switch plural boot programs without bothering an operator.

CONSTITUTION: Plural boot programs which boot a CPU 101 are stored in the ROM 104a to 104d respectively. The program stored in the ROM 104a is used for the ordinary booting operations. When a reset switch 106 is pushed, a reset part 102 resets the CPU 101 and also changes the address of the ROM to be asserted by an address decoder. Therefore, the internal address of the boot program contained in the CPU 101 and the external address of the actually corresponding ROM are converted. As a result, the boot program to be started is switched.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

[JP,08-179937,A]

*** NOTICES ***

Japan Patent Office is not responsible for any
damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] The boot program transfer device characterized by having a maintenance means to hold two or more boot programs for booting CPU, the resetting means which perform compulsive reset of Above CPU, and the change means which changes two or more boot programs held at the aforementioned maintenance means to the timing of reset by the aforementioned resetting means.

[Claim 2] It is the boot program transfer device according to claim 1 characterized by one being a fixed boot program usually used among two or more boot programs held at the aforementioned maintenance means.

[Claim 3] The aforementioned change means is a boot program transfer device according to claim 1 characterized by assigning the physical address in the aforementioned maintenance means to the logical address of the boot program in Above CPU.

[Claim 4] The aforementioned maintenance means is a boot program transfer device according to claim 1 characterized by being ROM.

[Claim 5] The aforementioned maintenance means is a boot program transfer device according to claim 4 which are two or more ROMs and is characterized by holding one boot program for every ROM.

[Claim 6] The aforementioned change means is a boot program transfer device according to claim 5 characterized by changing two or more boot programs by outputting the signal which specifies ***** of two or more aforementioned ROMs, and changing this signal.

[Claim 7] It is the boot program transfer device according to claim 2 which is equipped with the following and characterized by the aforementioned change means assigning the physical address in the aforementioned maintenance means to the logical address of the boot program in Above CPU one by one if the physical address of the aforementioned fixed boot program will be assigned to the logical address of the boot program in Above CPU if the fall of supply voltage is detected by the aforementioned voltage surveillance means, and Above CPU is reset from the aforementioned resetting means. Furthermore, an electric power supply means to supply power to Above CPU. A voltage surveillance means to supervise the supply voltage by the aforementioned electric power supply means.

[Claim 8] Furthermore, the boot program transfer device according to claim 1 characterized by resetting Above CPU by the aforementioned resetting means when it is judged that it has a time [to measure the elapsed time after Above CPU is reset] measurement means, and a judgment means to judge whether Above CPU was booted normally, and Above CPU was not normally booted by the aforementioned judgment means within the predetermined time.

[Claim 9] Furthermore, the boot program transfer device according to claim 1 characterized by resetting Above CPU by the aforementioned resetting means when it has a surveillance means to supervise the procedure in which Above CPU is booted normally and the aforementioned procedure is judged not to be equal to a predetermined procedure by the aforementioned surveillance means.

[Claim 10] Furthermore, the boot program transfer device according to claim 9 characterized by resetting Above CPU by the aforementioned resetting means when the procedure in which have a selection means to choose the aforementioned boot program, and Above CPU was normally booted by the aforementioned surveillance

means is judged not to be equal to the procedure of the boot by the boot program chosen with the aforementioned selection means.

[Claim 11] The boot program change method characterized by changing two or more aforementioned boot programs to the timing holding two or more boot programs for booting CPU by which forcible reset of the above CPU was carried out.

[Claim 12] It is the boot program change method according to claim 11 characterized by one being a fixed boot program usually used among two or more aforementioned boot programs.

[Claim 13] The boot program change method according to claim 11 characterized by changing two or more aforementioned boot programs by assigning the physical address in the aforementioned maintenance means to the logical address field of the boot program in Above CPU.

[Claim 14] Furthermore, the boot program change method according to claim 12 characterized by assigning the physical address in the aforementioned maintenance means to the logical address of the boot program in Above CPU one by one if the physical address of the aforementioned fixed boot program will be assigned to the logical address of the boot program in Above CPU if the electric power supply to Above CPU is supervised and the fall of the supply voltage to Above CPU is detected, and Above CPU is reset from the aforementioned resetting means.

[Claim 15] Furthermore, the boot program change method according to claim 11 characterized by resetting Above CPU by the aforementioned resetting means when it does not boot normally within a predetermined time, after Above CPU was reset.

[Claim 16] Furthermore, the boot program change method according to claim 11 characterized by resetting Above CPU by the aforementioned resetting means when the procedure in which Above CPU is booted normally is supervised and the aforementioned procedure is judged not to be equal to a predetermined procedure.

[Claim 17] Furthermore, the boot program change method according to claim 16 characterized by resetting Above CPU by the aforementioned resetting means when the procedure in which chose the aforementioned boot program and Above CPU was booted normally is judged not to be equal to the procedure of the boot by the boot program by which selection was carried out [aforementioned].

TECHNICAL PROBLEM

[Problem(s) to be Solved by the Invention] However, when the CPU system is equipped with two or more boot programs, the usual boot program is destroyed, and when it must boot by the boot program in another program ROM field and makes a boot program into the method of a change, it is necessary to change the map of the address bus of CPU on hardware. For that, the complicated operation of changing the switch on a substrate etc. is required, and possibility that a failure will occur in an unfamiliar operator especially is also high. Moreover, it becomes still more complicated operating it in using it, changing three or more sorts of boot programs.

[0004] It aims at offering automatically the boot program transfer device which can be changed, and its method, without being made in order that this invention may solve the technical problem mentioned above, and troubling an operator's hand for two or more boot programs.

[0015]

[Function] By the above composition, when the boot program of CPU needs to be changed, within equipment, the boot program which is outside the usual field automatically can be detected, and can be operated.

[0016]

[Example] Hereafter, one example concerning this invention is explained in detail with reference to a drawing.

[0017] It is drawing showing the composition of the boot program transfer device of CPU in this example in <1st example> drawing 1 . In addition, other composition, such as RAM which has extracted only the composition related to this example in drawing 1 , for example, serves as a working area of CPU, may be added.

[0018] In drawing 1, CPU101 is started from the boot program stored in Programs (ROM is called hereafter) 104a-ROMs 104d. The address of a boot program is connected by the low rank of an address signal 108 in ROMs 104a-104d. In addition, the address signal 108 shall show the address (boot address) of the boot program actually booted.

[0019] An address decoder 103 decodes the high order of an address signal 108, and generates chip select signals 112a-112d. The lot which forms any one [ROMs / 104a-104d] or the access unit of a data bus in case CPU101 accesses is selected by chip select signals 112a-112d. For example, ROM104a is selected by chip select signal 112a. In addition, in this example, the boot program which CPU101 shall usually start operation by the boot program in ROM104a for example, at the time of a power supply ON, and is hereafter stored in ROM104a is called a "fixed program."

[0020] The reset section 102 generates the reset pulse of predetermined width of face, when supply voltage and a reset switch 106 are supervised and the time of a supply voltage fall or a reset switch 106 is operated. This reset signal acts as the CPU reset signal 109 and address decoder reset signals 110 and 111 which control the chip select of an address decoder 103. The address decoder reset signal 110 is a reset signal generated by power supply surveillance, and the address decoder reset signal 111 is a reset signal at the time of operating a reset switch 106.

[0021] At the time of the usual power supply ON, the reset section 102 generates the CPU reset signal 109 and the address decoder reset signal 110. When the address decoder reset signal 110 occurs and CPU101 reboots, an address decoder 103 decodes the address signal 108 which CPU101 outputted as it is, and generates chip select signals 112a-112d. Here, the ROMs [104a-104d] address presupposes that the map is carried out to order from "0" in the order of ROM104a, ROM104b, ROM104c, and ROM104d. Then, usually, since CPU101 is accessed and boots from address = "0", immediately after boot, by asserting chip select signal 112a, namely, choosing ROM104a, a fixed program is started and the usual boot operation is performed.

[0022] On the other hand, when a reset switch 106 is operated, the reset section 102 generates the CPU reset signal 109 and the address decoder reset signal 111. This reboots CPU101 like the time of the power supply ON mentioned above. However, in this case, an address decoder 103 changes the address which CPU101 generates, and generates chip select signal 112 of ROM104b by which map is carried out to degree of usual boot ROM 104a b. Thereby, CPU101 tries operation by other boot programs other than the usual boot program.

[0023] As explained above, whenever the address decoder reset signal 111 occurs by operating a reset switch 106, an address decoder 103 changes the chip select signals 112a-112d of the boot address one by one, and goes.

[0024] In drawing 1, 105 is the normal boot Monitoring Department and is connected with the I/O Port of CPU101 etc. The normal boot Monitoring Department 105 supervises whether CPU101 accessed the normal boot Monitoring Department 105 by software operation of a predetermined procedure immediately after boot of CPU101. And it reports to an operator whether CPU101 booted normally by operating flicker of display Light Emitting Diode 107 corresponding to it. An operator will do the depression of the reset switch 106, if it recognizes that CPU101 did not boot normally. Then, an address decoder 103 changes a boot program field (ROMs 104a-104d) by changing chip select signals 112a-112d, as mentioned above corresponding to operation of a reset switch. That is, a ROMs [104a-104d] physical address is assigned to the logical address of the boot program in CPU101 one by one.

[0025] Hereafter, the detailed composition of the reset section 102 and an address decoder 103 is shown in drawing 2. In drawing 2, the same number is given to the same component as drawing 1 mentioned above, and explanation is omitted.

[0026] In drawing 2, the reset section 102 consists of the power supply Monitoring Department 201, the switch reset section 202, and the OR gate 203. The power supply Monitoring Department 201 generates the negative reset pulse of a predetermined time as an address decoder reset signal 110, when the time of a power supply ON and supply voltage fall from a predetermined value. Similarly, the reset switch section 202 shall generate the negative reset pulse of a predetermined time as an address decoder reset signal 111, when a reset switch 106 is

operated. The OR gate 203 takes the OR of the address decoder reset signal 110,111, and generates the reset signal 109 which resets CPU101.

[0027] On the other hand, an address decoder 103 consists of a counter 204, a decoder 205, and the gate section 206. Here, a counter 204 has the counted value of 2 bits, and a decoder 205 assumes that it is a 2-bit decoder. A decoder 205 asserts either of the 4-bit outputs according to the output of a counter 204. Moreover, the gate section 206 asserts chip select signals [112a-112d] either according to the address signal 108 inputted from CPU101.

[0028] The timing chart of each signal in drawing 2 is shown in drawing 3 . Suppose that the power supply was changed from OFF to ON in the zero in the timing chart shown in drawing 3 . Then, as for the power supply surveillance reset signal 110, only a predetermined time generates a negative pulse from the time of supply voltage going up and reaching predetermined voltage. Then, CPU101 is booted and starts operation from the predetermined program address. In addition, the chip select signal of ROM104a which the predetermined program address is the address "0" and is usually equivalent to the address "0" is 112a.

[0029] A reset signal 110 clears a counter 204, sets the output signal 207 from a counter 204 to "0", and asserts 208a among the output signals of a decoder 205.

[0030] When CPU101 does not boot immediately after a power supply ON and within a predetermined time, an operator detects it by Light Emitting Diode107, and operates a reset switch 106. Then, the switch reset section 202 generates the negative pulse of predetermined time width of face in the address decoder reset signal 111, and this is inputted into the clock of a counter 204 and counts up a counter 204 while it boots CPU101 through a reset signal 109.

[0031] According to the output 207 counted up in the counter 204, a decoder 205 asserts the output. That is, as shown in drawing 3 , when the counter output 207 is "0", a decoder 205 asserts output 208a, and when the counter output 207 is "1", output 208b is asserted. Similarly, 208d will be asserted, if the counter output 207 is "3" and it is output 208c and "4."

[0032] And in the gate section 206, chip select signals [112a-112d] either is asserted according to the output 207 from the address signal 108 inputted from CPU101, and a counter 204. For example, if output 208a of a decoder 205 is "1", when an address signal 108 is "0", chip select signal 112a is asserted as an output from the gate section 206. Moreover, when an address signal 108 is "1" and chip select signal 112b is [address signals 108] "2" and "3" at this time, chip select signals 112c and 112d are asserted, respectively. This is equivalent to normal ROM access in the state of normal boot.

[0033] Moreover, when output 208b of a decoder 205 is "1", chip select signal 112b is asserted in the state of "0" at the time 108 of boot, i.e., an address signal. When output 208c of a decoder 205 or 208d are "1" similarly, an address signal 108 asserts chip select signals 112c and 112d at the time of boot of "0."

[0034] Thus, according to the output 207 counted in the counter 204, it changes to chip select signals 112b, 112c, and 112d one by one sequentially from chip select signal 112a corresponding to the address "0."

[0035] The storing address of a boot program can be changed one by one until CPU is booted in a predetermined time according to this example, as explained above. Therefore, starting from other boot programs can be performed, without performing complicated operation.

[0036] In addition, although this example explained the case where the fixed boot program was stored in ROM104a, this invention is good also considering the boot program which is not restricted to this example and stored in which ROM as a fixed program. In this case, what is necessary is just to change the composition of the gate section 206 shown in drawing 2 so that the chip select signal asserted when an address signal 108 is "0" may become in addition to 112a.

[0037] Moreover, although the case where ROM which stores a boot program was plurality was explained, a boot program may be stored in two or more addresses different, for example within single ROM, and you may constitute so that a physical address map in case the address signal 108 from CPU is "0" may be changed according to the counter output 207.

[0038] The 2nd example concerning this invention is explained below the <2nd example>.

[0039] The composition of the boot program transfer device of CPU in the 2nd example is shown in drawing 4 . In drawing 4 , the same number is attached about the same composition as drawing 1 in the 1st example mentioned above, and explanation is omitted.

[0040] In the 2nd example, CPU401 builds in the watchdog timer 403. Here, a watchdog timer 403 is a timer which operates with the clock of CPU401. A watchdog timer 403 outputs a reset signal 111 while it judges it as a CPU overrun and resets operation of CPU401, when reset by software operation is not performed within a predetermined time.

[0041] Although the reset section 402 supervises supply voltage and a reset switch 106 like the reset section 102 shown in drawing 2 mentioned above, it is characterized by having not separated both output.

[0042] In the 2nd example, it clears the counter 204 of the address decoder 103 interior while making CPU401 reset and boot like the 1st example mentioned above immediately after a power supply ON.

[0043] Moreover, a watchdog timer 403 generates a reset signal 111, and counts up the counter 204 of an address decoder 103 while it resets CPU401, when not reset in a predetermined time. And when it does not boot normally by ROM selected as a result, a watchdog timer 403 continues outputting a reset signal 111 at intervals of a predetermined time, and an address decoder 103 changes selection of ROMs 104a-104d, and it detects automatically the program ROM booted normally.

[0044] If it cannot boot after predetermined-time progress according to the 2nd example even if there is no depression of the reset switch by the operator as explained above, it can change to the address in which the following boot program is stored automatically.

[0045] The 3rd example concerning this invention is explained below the <3rd example>.

[0046] The composition of the boot address translation section of CPU in the 3rd example is shown in drawing 5 . In drawing 5 , the same number is attached about the same composition as drawing 1 in the 1st example mentioned above, and explanation is omitted.

[0047] In the 3rd example, it is characterized by having the effect which combined the 1st example and the 2nd example which were mentioned above.

[0048] In drawing 5 , the normal boot Monitoring Department 501 supervises access in the predetermined procedure of CPU101 within the predetermined time immediately after boot of CPU101 with a timer 502. And from CPU101, when there is no access with a predetermined procedure, it judges that it is a CPU overrun, and a reset signal is generated. This reset signal is compounded with the signal from a reset switch 106 at the gate 503, and is inputted into the reset section 102.

[0049] In the reset section 102, whenever the normal boot Monitoring Department 501 generates a reset signal, a reset signal 111 is generated, and the counter 204 of the address decoder 103 interior is counted up. This searches the boot program booted normally with the time interval set up with the timer 502 like the 2nd example mentioned above.

[0050] If it cannot boot after predetermined-time progress according to the 3rd example even if there is a depression of the reset switch by the operator or there is no depression of a reset switch by combining the 1st example and the 2nd example which were mentioned above as explained above, it can change to the address in which the following boot program is stored.

[0051] The 4th example concerning this invention is explained below the <4th example>.

[0052] The composition of the boot program transfer device of CPU in the 4th example is shown in drawing 6 . In drawing 6 , the same number is attached about the same composition as drawing 1 in the 1st example mentioned above, and explanation is omitted.

[0053] In drawing 6 , it is the boot program selecting switch 601, and an operator can choose a boot program. the access method to the boot program chosen by the boot program selecting switch 601 is boiled in the normal boot Monitoring Department 501, and it has it as an access pattern table beforehand Although the normal boot Monitoring Department 501 detects the existence of access of a predetermined procedure from CPU101 within an after [boot] predetermined time like the 3rd example mentioned above, whether it is the access pattern with which the access method from CPU101 was chosen by the boot program selecting switch 601 detects it

further.

[0054] When it is judged at the normal boot Monitoring Department 501 that normal access was not performed from CPU101 (for example, when different access from the access method chosen by the boot program selecting switch 601 is performed), the normal boot Monitoring Department 501 chooses ROM by which the following boot program is stored in the OR gate 503 like the 3rd example which generated the reset signal and was mentioned above.

[0055] As explained above, according to the 4th example, arbitrary boot programs can be easily operated by memorizing the access pattern which discriminates it for every boot program beforehand.

[0056] In addition, in the 1st ~ the 4th example which were mentioned above, although the example the number of boot programs is [example] four was explained, this invention can be applied, when not this limitation but two or more boot programs exist, of course.

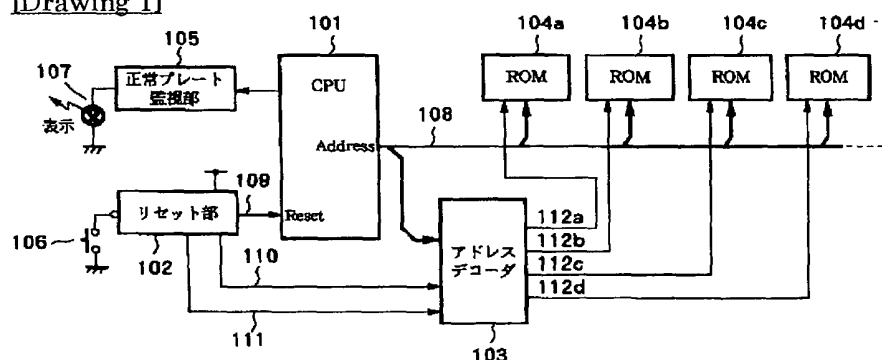
[0057] In addition, even if it applies this invention to the system which consists of two or more devices, you may apply it to the equipment which consists of one device. Moreover, this invention cannot be overemphasized by that it can apply when attained by supplying a program to a system or equipment.

[0058]

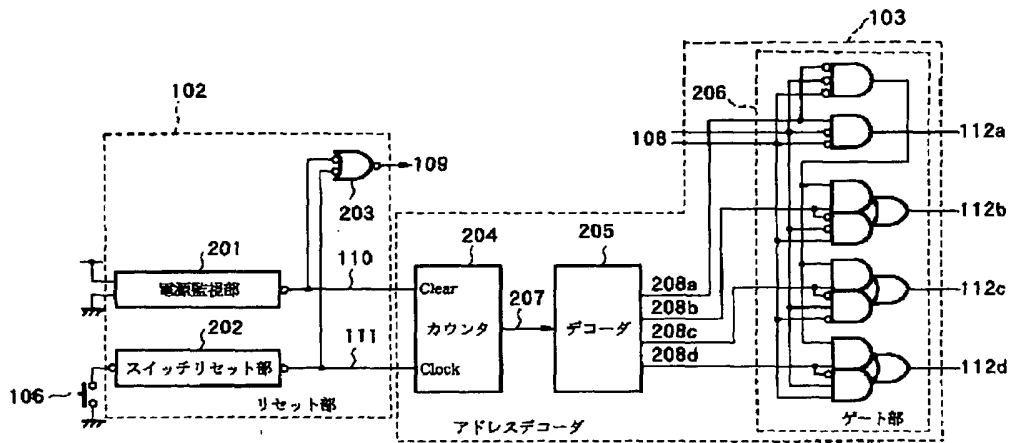
[Effect of the Invention] It becomes possible to change a boot program field, without performing complicated operation in a CPU system with two or more boot program fields according to this invention, as explained above.

DRAWINGS

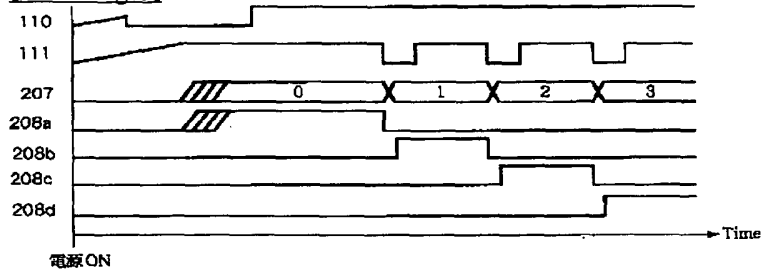
[Drawing 1]



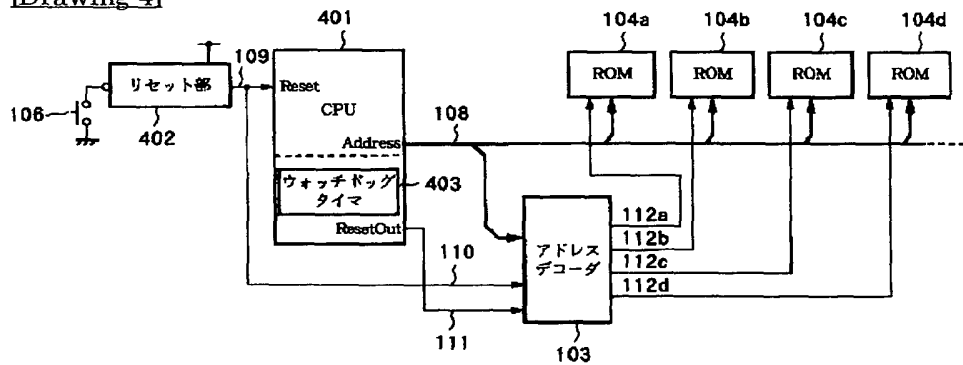
[Drawing 2]



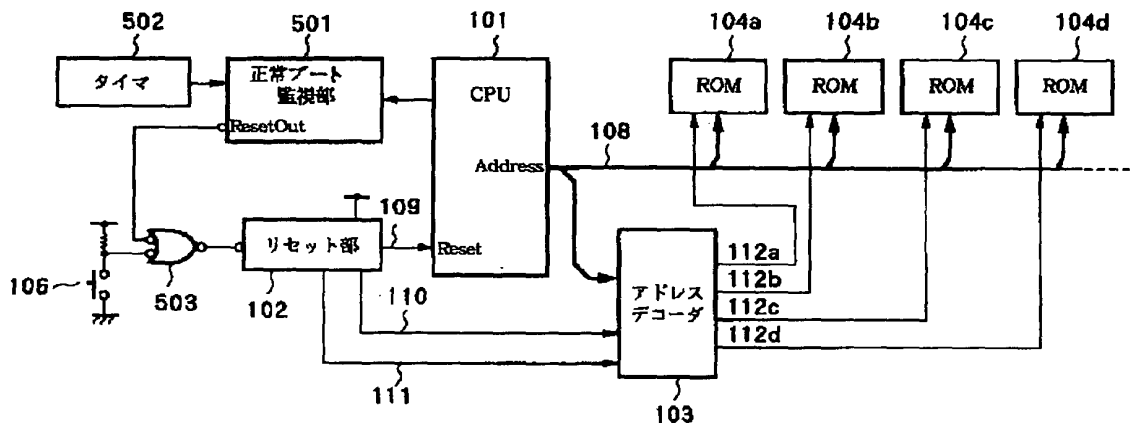
[Drawing 3]



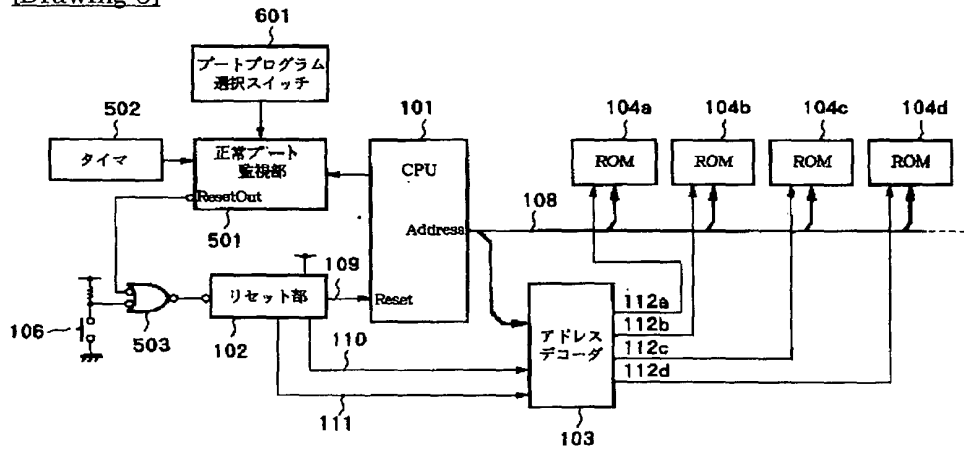
[Drawing 4]



[Drawing 5]



[Drawing 6]



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-179937

(43) 公開日 平成8年(1996)7月12日

(51) Int.Cl.⁶

G 0 6 F 9/08
11/14

識別記号

4 1 0 V
3 1 0 L

庁内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数17 O L (全 8 頁)

(21) 出願番号 特願平6-324423

(22) 出願日 平成6年(1994)12月27日

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72) 発明者 本間 英雄

東京都大田区下丸子3丁目30番2号 キヤ
ノン株式会社内

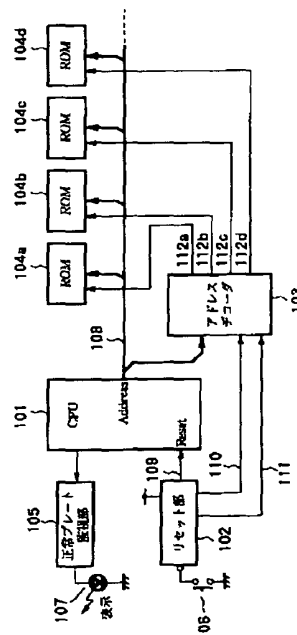
(74) 代理人 弁理士 大塚 康徳 (外1名)

(54) 【発明の名称】 ブートプログラム切替え装置及びその方法

(57) 【要約】

【目的】 複数のブートプログラムを操作者の手を煩わせることなく、自動的に切替可能なブートプログラム切替え装置及びその方法を提供することを目的とする。

【構成】 CPU101をブートする複数のブートプログラムが、ROM104a～104dにそれぞれ格納されており、ROM104aに格納されているものが通常ブート用である。リセットスイッチ106が押下されると、リセット部102はCPU101をリセットするとともに、アドレスデコーダがアサートするROMのアドレスを変更する。従って、CPU101の有するブートプログラムの内部アドレスと、実際に対応するROM上の外部アドレスとが変換され、結果として起動されるブートプログラムを切り替えることができる。



【特許請求の範囲】

【請求項 1】 CPU をブートするためのブートプログラムを複数保持する保持手段と、

前記 CPU の強制リセットを行うリセット手段と、

前記リセット手段によるリセットのタイミングで、前記保持手段に保持された複数のブートプログラムを切り替える切替え手段とを有することを特徴とするブートプログラム切替え装置。

【請求項 2】 前記保持手段に保持された複数のブートプログラムのうち、1 つは通常使用される既定ブートプログラムであることを特徴とする請求項 1 記載のブートプログラム切替え装置。

【請求項 3】 前記切替え手段は、前記 CPU におけるブートプログラムの論理アドレスに前記保持手段における物理アドレスを割り当てることを特徴とする請求項 1 記載のブートプログラム切替え装置。

【請求項 4】 前記保持手段は ROM であることを特徴とする請求項 1 記載のブートプログラム切替え装置。

【請求項 5】 前記保持手段は複数の ROM であり、各 ROM 毎に 1 つのブートプログラムを保持することを特徴とする請求項 4 記載のブートプログラム切替え装置。

【請求項 6】 前記切替え手段は、前記複数の ROM のいずれかを特定する信号を出力し、該信号を切り替えることにより複数のブートプログラムを切り替えることを特徴とする請求項 5 記載のブートプログラム切替え装置。

【請求項 7】 更に、前記 CPU に電力を供給する電力供給手段と、

前記電力供給手段による供給電圧を監視する電圧監視手段とを有し、

前記切替え手段は、前記電圧監視手段により供給電圧の低下が検出されると前記 CPU におけるブートプログラムの論理アドレスに前記既定ブートプログラムの物理アドレスを割り当て、前記リセット手段より前記 CPU がリセットされると前記 CPU におけるブートプログラムの論理アドレスに前記保持手段における物理アドレスを順次割り当てることを特徴とする請求項 2 記載のブートプログラム切替え装置。

【請求項 8】 更に、前記 CPU がリセットされてからの経過時間を計測する時間計測手段と、

前記 CPU が正常にブートされたか否かを判断する判断手段とを有し、

前記判断手段により前記 CPU が所定時間以内に正常にブートされなかったと判断された場合、前記リセット手段により前記 CPU をリセットすることを特徴とする請求項 1 記載のブートプログラム切替え装置。

【請求項 9】 更に、前記 CPU が正常にブートされる手順を監視する監視手段を有し、

前記監視手段により前記手順が所定の手順に等しくない

と判断された場合、前記リセット手段により前記 CPU をリセットすることを特徴とする請求項 1 記載のブートプログラム切替え装置。

【請求項 10】 更に、前記ブートプログラムを選択する選択手段を有し、

前記監視手段により前記 CPU が正常にブートされた手順が前記選択手段で選択されたブートプログラムによるブートの手順に等しくないと判断された場合、前記リセット手段により前記 CPU をリセットすることを特徴とする請求項 9 記載のブートプログラム切替え装置。

【請求項 11】 CPU をブートするためのブートプログラムを複数保持し、

前記 CPU が強制リセットされたタイミングで、前記複数のブートプログラムを切り替えることを特徴とするブートプログラム切替え方法。

【請求項 12】 前記複数のブートプログラムのうち、1 つは通常使用される既定ブートプログラムであることを特徴とする請求項 11 記載のブートプログラム切替え方法。

【請求項 13】 前記 CPU におけるブートプログラムの論理アドレス領域に前記保持手段における物理アドレスを割り当てることにより、前記複数のブートプログラムを切り替えることを特徴とする請求項 11 記載のブートプログラム切替え方法。

【請求項 14】 更に、前記 CPU への電力供給を監視し、

前記 CPU への供給電圧の低下が検出されると前記 CPU におけるブートプログラムの論理アドレスに前記既定ブートプログラムの物理アドレスを割り当て、前記リセット手段より前記 CPU がリセットされると前記 CPU におけるブートプログラムの論理アドレスに前記保持手段における物理アドレスを順次割り当てることを特徴とする請求項 12 記載のブートプログラム切替え方法。

【請求項 15】 更に、前記 CPU がリセットされてから所定時間以内に正常にブートされなかった場合、前記リセット手段により前記 CPU をリセットすることを特徴とする請求項 11 記載のブートプログラム切替え方法。

【請求項 16】 更に、前記 CPU が正常にブートされる手順を監視し、

前記手順が所定の手順に等しくないと判断された場合、前記リセット手段により前記 CPU をリセットすることを特徴とする請求項 11 記載のブートプログラム切替え方法。

【請求項 17】 更に、前記ブートプログラムを選択し、

前記 CPU が正常にブートされた手順が前記選択されたブートプログラムによるブートの手順に等しくないと判断された場合、前記リセット手段により前記 CPU をリセットすることを特徴とする請求項 16 記載のブート

ログラム切替え方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明はブートプログラム切替え装置及びその方法に関し、例えば複数のブートプログラム領域を備えるCPUのブートプログラム切替え装置及びその方法に関する。

【0002】

【従来の技術】従来のCPUを備えるシステムにおけるブートプログラムは、プログラムROM領域内で、CPUのブートアドレスとして固定された所定の箇所に

【0003】

【発明が解決しようとする課題】しかしながら、CPUシステムがブートプログラムを複数備えている場合に、例えば通常のブートプログラムが破壊され、別のプログラムROM領域にあるブートプログラムでブートしなければならぬ時等、ブートプログラムを切り替えようとする場合には、CPUのアドレスバスのマップをハードウェア上で切り替える必要がある。このためには、基板

上のスイッチを切り替える等、煩雑な操作が必要であり、特に不慣れなオペレータでは操作ミスが発生する可能性も高い。また、3種以上のブートプログラムを切り替えて使用するような場合には、なおさら操作が繁雑となる。

【0004】本発明は上述した課題を解決するためにな

されたものであり、複数のブートプログラムを操作者の手を煩わせることなく、自動的に切替可能なブートプログラム切替え装置及びその方法を提供することを目的とする。

【0005】

【課題を解決するための手段】上述した目的を達成するための一手段として、本発明は以下の構成を備える。即ち、CPUをブートするためのブートプログラムを複数

保持する保持手段と、前記CPUの強制リセットを行うリセット手段と、前記リセット手段によるリセットのタイミングで、前記保持手段に保持された複数のブートプログラムを切り替える切替え手段とを有することを特徴とする。

【0006】例えば、前記保持手段に保持された複数のブートプログラムのうち、1つは通常使用される既定ブートプログラムであることを特徴とする。

【0007】例えば、前記切替え手段は、前記CPUにおけるブートプログラムの論理アドレス領域に前記保持手段における物理アドレスを割り当てることを特徴とする。

【0008】例えば、前記保持手段はROMであることを特徴とする。

【0009】例えば、前記保持手段は複数のROMであり、各ROM毎に1つのブートプログラムを保持することを特徴とする。

【0010】例えば、前記切替え手段は、前記複数のROMのいずれかを特定する信号を出力し、該信号を切り替えることにより複数のブートプログラムを切り替えることを特徴とする。

【0011】更に、前記CPUに電力を供給する電力供給手段と、前記電力供給手段による供給電圧を監視する電圧監視手段とを有し、前記切替え手段は、前記電圧監視手段により供給電圧の低下が検出されると前記CPUにおけるブートプログラムの論理アドレスに前記既定ブートプログラムの物理アドレスを割り当て、前記リセット手段より前記CPUがリセットされると前記CPUにおけるブートプログラムの論理アドレスに前記保持手段における物理アドレスを順次割り当てることを特徴とする。

【0012】更に、前記CPUがリセットされてからの経過時間を計測する時間計測手段と、前記CPUが正常にブートされたか否かを判断する判断手段とを有し、前記判断手段により前記CPUが所定時間以内に正常にブートされなかったと判断された場合、前記リセット手段により前記CPUをリセットすることを特徴とする。

【0013】更に、前記CPUが正常にブートされる手順を監視する監視手段を有し、前記監視手段により前記手順が所定の手順に等しくないと判断された場合、前記リセット手段により前記CPUをリセットすることを特徴とする。

【0014】更に、前記ブートプログラムを選択する選択手段を有し、前記監視手段により前記CPUが正常にブートされた手順が前記選択手段で選択されたブートプログラムによるブートの手順に等しくないと判断された場合、前記リセット手段により前記CPUをリセットすることを特徴とする。

【0015】

【作用】以上の構成により、CPUのブートプログラムを切り替える必要が生じた場合に、装置内で自動的に通常の領域外にあるブートプログラムを検出し、動作させる事ができる。

【0016】

【実施例】以下、本発明に係る一実施例について、図面を参照して詳細に説明する。

【0017】<第1実施例>図1に、本実施例におけるCPUのブートプログラム切替え装置の構成を示す図である。尚、図1においては本実施例に係る構成のみを抜粋しており、例えばCPUの作業領域となるRAM等の他の構成が追加されていても良い。

【0018】図1において、CPU101はプログラムROM（以下、ROMと称する）104a～104dに格納されたブートプログラムより起動される。ROM1

04 a~104 dにおいて、ブートプログラムのアドレスはアドレス信号108の下位により接続される。尚、アドレス信号108は、実際にブートされるブートプログラムのアドレス（ブートアドレス）を示しているものとする。

【0019】アドレスデコーダ103は、アドレス信号108の上位をデコードし、チップセレクト信号112 a~112 dを生成する。CPU101がアクセスする際にROM104 a~104 dのいずれか1つ、あるいはデータバスのアクセス単位を形成する一組が、チップセレクト信号112 a~112 dでセレクトされる。例えば、チップセレクト信号112 aによりROM104 aがセレクトされる。尚、本実施例においては、通常、例えば電源ON時にはROM104 aにあるブートプログラムによりCPU101は動作を開始するものとし、以下、ROM104 aに格納されているブートプログラムを「既定プログラム」と称する。

【0020】リセット部102は、電源電圧及びリセットスイッチ106を監視して、電源電圧低下時あるいはリセットスイッチ106が操作された場合に、所定の幅のリセットパルスを発生する。このリセット信号はCPUリセット信号109、及びアドレスデコーダ103のチップセレクトを制御するアドレスデコーダリセット信号110、111として作用する。アドレスデコーダリセット信号110は電源監視により発生するリセット信号であり、アドレスデコーダリセット信号111はリセットスイッチ106を動作させた場合のリセット信号である。

【0021】通常の電源ON時には、リセット部102はCPUリセット信号109及びアドレスデコーダリセット信号110を発生する。アドレスデコーダリセット信号110が発生し、CPU101がリポートした場合に、アドレスデコーダ103はCPU101の出力したアドレス信号108をそのままデコードしてチップセレクト信号112 a~112 dを発生する。ここで、ROM104 a~104 dのアドレスが、ROM104 a、ROM104 b、ROM104 c、ROM104 dの順で「0」から順にマップされているとする。すると、通常CPU101はアドレス＝「0」からアクセスしてブートするため、ブート直後はチップセレクト信号112 aがアサートされ、即ちROM104 aが選択されることにより既定プログラムが起動され、通常のブート動作が行われる。

【0022】一方、リセットスイッチ106が操作されたときには、リセット部102はCPUリセット信号109及びアドレスデコーダリセット信号111を発生する。これによりCPU101は上述した電源ON時と同様にリポートする。しかしながら、この場合アドレスデコーダ103はCPU101が発生するアドレスを変換して、通常のブートROM104 aの次にマップされて

いるROM104 bのチップセレクト信号112 bを発生する。これにより、CPU101は通常のブートプログラム以外の他のブートプログラムによる動作を試みる。

【0023】以上説明したように、リセットスイッチ106を操作することによりアドレスデコーダリセット信号111が発生する毎に、アドレスデコーダ103はブートアドレスのチップセレクト信号112 a~112 dを順次切り替えて行く。

【0024】図1において105は正常ブート監視部であり、CPU101のI/Oポート等と接続されている。正常ブート監視部105は、CPU101のブート直後に、CPU101が所定の手順のソフトウェア操作で正常ブート監視部105にアクセスしたか否かを監視する。そして、それに対応して表示LED107の明滅を操作することにより、CPU101が正常にブートしたか否かを操作者に報知する。操作者はCPU101が正常にブートしなかったことを認識すると、リセットスイッチ106を押下する。すると、アドレスデコーダ103はリセットスイッチの操作に対応して上述したようにチップセレクト信号112 a~112 dを切り替えることにより、ブートプログラム領域（ROM104 a~104 d）を切り替える。即ち、CPU101におけるブートプログラムの論理アドレスに、ROM104 a~104 dの物理アドレスを順次割り当てる。

【0025】以下、図2にリセット部102及びアドレスデコーダ103の詳細構成を示す。図2において、上述した図1と同一の構成要素には同一番号を付し、説明を省略する。

【0026】図2において、リセット部102は電源監視部201とスイッチリセット部202、及びORゲート203とから成る。電源監視部201は、電源ON時及び電源電圧が所定値より下がった場合に、所定時間の負のリセットパルスをアドレスデコーダリセット信号110として発生する。同様に、リセットスイッチ部202はリセットスイッチ106が操作された場合に、所定時間の負のリセットパルスをアドレスデコーダリセット信号111として発生するものとする。ORゲート203は、アドレスデコーダリセット信号110、111の論理和をとり、CPU101をリセットするリセット信号109を発生する。

【0027】一方、アドレスデコーダ103は、カウンタ204、デコーダ205、ゲート部206とから成る。ここで、カウンタ204は2ビットのカウント値を有し、デコーダ205は2ビットのデコーダであると仮定する。デコーダ205は、カウンタ204の出力に応じて4ビット出力のいずれかをアサートする。また、ゲート部206はCPU101から入力されるアドレス信号108に応じて、チップセレクト信号112 a~112 dのいずれかをアサートする。

【0028】図3に、図2における各信号のタイミングチャートを示す。図3に示すタイミングチャートでは、原点において電源をOFFからONに切り替えたとする。すると、電源監視リセット信号110は、電源電圧が上昇して所定電圧に達した時点から、所定時間だけ負のバルスが発生する。この後、CPU101はブートし、所定のプログラムアドレスから動作を開始する。尚、通常、所定のプログラムアドレスはアドレス「0」であり、アドレス「0」に相当するROM104aのチップセレクト信号は112aである。

【0029】リセット信号110は、カウンタ204をクリアして、カウンタ204からの出力信号207を「0」にして、デコーダ205の出力信号のうち、208aをアサートする。

【0030】電源ONの直後、所定時間以内にCPU101がブートしない場合、操作者はLED107でそれを検出し、リセットスイッチ106を操作する。すると、スイッチリセット部202はアドレスデコーダリセット信号111に所定の時間幅の負のバルスが発生し、これはリセット信号109を介してCPU101をブートするとともに、カウンタ204のクロックに入力されてカウンタ204をカウントアップする。

【0031】カウンタ204においてカウントアップされた出力207に応じて、デコーダ205はその出力をアサートする。即ち、図3に示すように、カウンタ出力207が「0」である場合にはデコーダ205は出力208aをアサートし、カウンタ出力207が「1」である場合には出力208bをアサートする。同様に、カウンタ出力207が「3」であれば出力208c、「4」であれば208dがアサートされる。

【0032】そして、ゲート部206において、CPU101から入力されるアドレス信号108及びカウンタ204からの出力207に応じて、チップセレクト信号112a～112dのいずれかがアサートされる。例えば、デコーダ205の出力208aが「1」であれば、アドレス信号108が「0」である場合にはゲート部206からの出力としてチップセレクト信号112aがアサートされる。またこのとき、アドレス信号108が「1」である場合にはチップセレクト信号112bが、アドレス信号108が「2」及び「3」である場合には、それぞれチップセレクト信号112c、112dがアサートされる。これは正常ブート状態で正常なROMアクセスに相当する。

【0033】また、デコーダ205の出力208bが「1」である場合はブート時、即ちアドレス信号108が「0」の状態ではチップセレクト信号112bをアサートする。同様にデコーダ205の出力208cもしくは208dが「1」である場合には、アドレス信号108が「0」のブート時にチップセレクト信号112c、112dをアサートする。

【0034】このように、カウンタ204においてカウントされた出力207に応じて、アドレス「0」に対応したチップセレクト信号112aから順に、チップセレクト信号112b、112c、112dと、順次切り替えられる。

【0035】以上説明した様に本実施例によれば、CPUが所定時間内にブートされるまでブートプログラムの格納アドレスを順次変更していくことができる。従って、煩雑な操作を行うことなく、他のブートプログラムからの起動が行える。

【0036】尚、本実施例では既定ブートプログラムがROM104aに格納されている場合について説明を行ったが、本発明はこの例に限られるものではなく、どのROMに格納されているブートプログラムを既定プログラムとしてもよい。この場合、図2に示すゲート部206の構成を、アドレス信号108が「0」である場合にアサートされるチップセレクト信号が112a以外になるように変更すれば良い。

【0037】また、ブートプログラムを格納するROMが複数である場合について説明を行ったが、例えば単一のROM内で異なる複数のアドレスにブートプログラムを格納し、カウンタ出力207に応じてCPUからのアドレス信号108が「0」である場合の物理アドレスマップを変更するように構成してもよい。

【0038】＜第2実施例＞以下、本発明に係る第2実施例について説明する。

【0039】図4に、第2実施例におけるCPUのブートプログラム切替え装置の構成を示す。図4において、上述した第1実施例における図1と同様の構成については同一番号を付し、説明を省略する。

【0040】第2実施例において、CPU401はウォッチドッグタイマ403を内蔵している。ここで、ウォッチドッグタイマ403は、CPU401のクロックで動作するタイマである。ウォッチドッグタイマ403は、所定時間以内にソフトウェア操作によるリセットが行なわれない場合には、CPU暴走と判断してCPU401の動作をリセットするとともに、リセット信号111を出力する。

【0041】リセット部402は、上述した図2に示したリセット部102と同様に電源電圧及びリセットスイッチ106を監視するが、両者の出力は分離していないことを特徴とする。

【0042】第2実施例においては、電源ON直後は上述した第1実施例と同様にCPU401をリセットしてブートさせるとともに、アドレスデコーダ103内部のカウンタ204をクリアする。

【0043】また、ウォッチドッグタイマ403は、所定時間内にリセットされない場合にCPU401をリセットするとともに、リセット信号111を発生させ、アドレスデコーダ103のカウンタ204をカウントアップ

ブする。そして、その結果セレクトされたROMでも正常にブートしなかった場合、ウォッチドッグタイマ403は所定時間間隔でリセット信号111を出力し続け、アドレスデコーダ103はROM104a~104dの選択を切り替えてゆき、正常にブートするプログラムROMを自動的に検出する。

【0044】以上説明したように第2実施例によれば、操作者によるリセットスイッチの押下が無くても、所定時間経過後にブートできなければ、自動的に次のブートプログラムが格納されているアドレスに切り替えることができる。

【0045】<第3実施例>以下、本発明に係る第3実施例について説明する。

【0046】図5に、第3実施例におけるCPUのブートアドレス変換部の構成を示す。図5において、上述した第1実施例における図1と同様の構成については同一番号を付し、説明を省略する。

【0047】第3実施例においては、上述した第1実施例及び第2実施例とを組み合わせた効果を有することを特徴とする。

【0048】図5において、正常ブート監視部501はタイマ502でCPU101のブート直後の所定時間以内に、CPU101からの所定の手順でアクセスを監視する。そして、CPU101から所定の手順でのアクセスがない場合にはCPU暴走であると判断し、リセット信号を発生する。該リセット信号はゲート503でリセットスイッチ106からの信号と合成され、リセット部102に入力される。

【0049】リセット部102では、正常ブート監視部501がリセット信号を発生させる毎にリセット信号111を発生させ、アドレスデコーダ103内部のカウンタ204をカウントアップする。これにより、上述した第2実施例と同様に、タイマ502で設定された時間間隔で正常にブートするブートプログラムを検索する。

【0050】以上説明したように第3実施例によれば、上述した第1実施例及び第2実施例を組み合わせることにより、操作者によるリセットスイッチの押下があるか、またはリセットスイッチの押下が無くても所定時間経過後にブートできなければ、次のブートプログラムが格納されているアドレスに切り替えることができる。

【0051】<第4実施例>以下、本発明に係る第4実施例について説明する。

【0052】図6に、第4実施例におけるCPUのブートプログラム切替え装置の構成を示す。図6において、上述した第1実施例における図1と同様の構成については同一番号を付し、説明を省略する。

【0053】図6においては、ブートプログラム選択スイッチ601で、操作者がブートプログラムを選択することができる。ブートプログラム選択スイッチ601で選択されるブートプログラムへのアクセス手順は、正常

ブート監視部501内に、予めアクセスパターンテーブルとして備えられている。正常ブート監視部501は、上述した第3実施例と同様にブート後所定時間以内にCPU101から所定手順のアクセスの有無を検出するが、更に、CPU101からのアクセス手順がブートプログラム選択スイッチ601で選択されたアクセスパターンであるか否かも検出する。

【0054】正常ブート監視部501において、CPU101から正常なアクセスが行われなかったと判断される場合、例えば、ブートプログラム選択スイッチ601で選択されたアクセス手順と異なるアクセスが行われた場合、正常ブート監視部501はORゲート503へリセット信号を発生し、上述した第3実施例同様に、次のブートプログラムが格納されているROMを選択する。

【0055】以上説明したように第4実施例によれば、予めブートプログラム毎にそれを識別するアクセスパターンを記憶しておくことにより、容易に任意のブートプログラムを動作させることができる。

【0056】尚、上述した第1~第4実施例においては、ブートプログラムが4つである例について説明を行ったが、本発明はもちろんこの限りではなく、2つ以上のブートプログラムが存在する場合に適用可能である。

【0057】尚、本発明は、複数の機器から構成されるシステムに適用しても、1つの機器から成る装置に適用しても良い。また、本発明はシステム或は装置にプログラムを供給することによって達成される場合にも適用できることはいうまでもない。

【0058】

【発明の効果】以上説明したように本発明によれば、複数のブートプログラム領域を持つCPUシステムにおいて、複雑な操作を行うことなくブートプログラム領域を変更することが可能となる。

【0059】

【図面の簡単な説明】

【図1】本発明に係る一実施例におけるCPUのブートプログラム切替え装置の構成を示すブロック図である。

【図2】本実施例におけるリセット部及びアドレスデコーダの詳細構成を示す図である。

【図3】本実施例のリセット部及びアドレスデコーダにおけるタイミングチャートである。

【図4】本発明に係る第2実施例におけるCPUのブートプログラム切替え装置の構成を示すブロック図である。

【図5】本発明に係る第3実施例におけるCPUのブートプログラム切替え装置の構成を示すブロック図である。

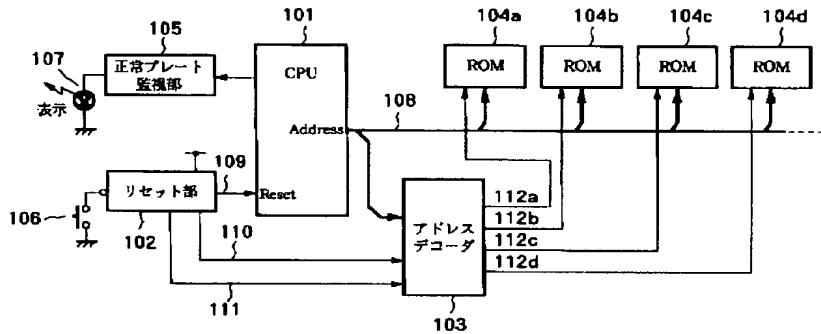
【図6】本発明に係る第4実施例におけるCPUのブートプログラム切替え装置の構成を示すブロック図である。

【符号の説明】

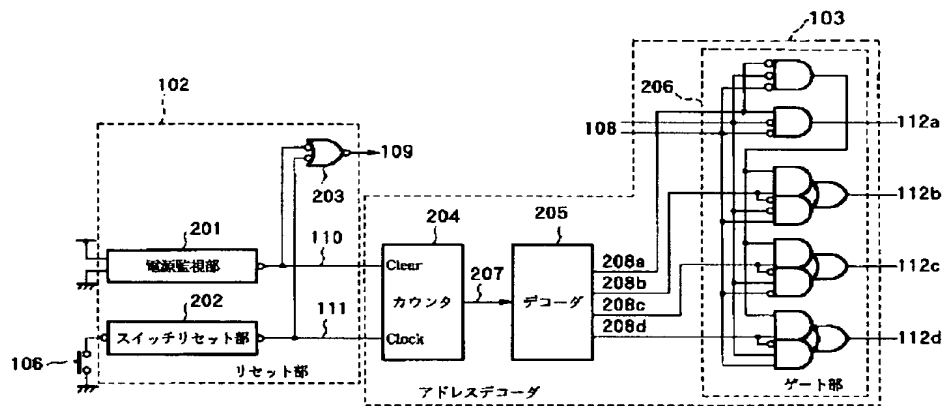
11
104, 401 CPU
102 リセット部
103 アドレスデコーダ
104a~104d ROM
105, 501 正常ブート監視部
106, 402 リセットスイッチ、

12
* 201 電源監視リセット部
202 スイッチリセット部
204 カウンタ
205 デコーダ
206 ゲート部
* 601 ブートプログラム選択スイッチ

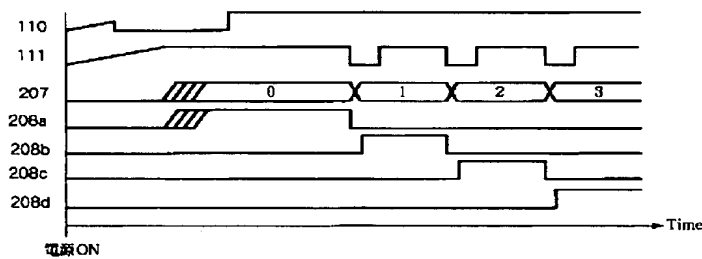
【図 1】



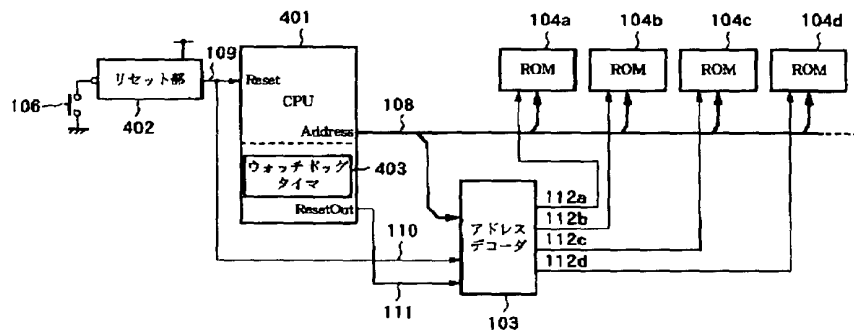
【図 2】



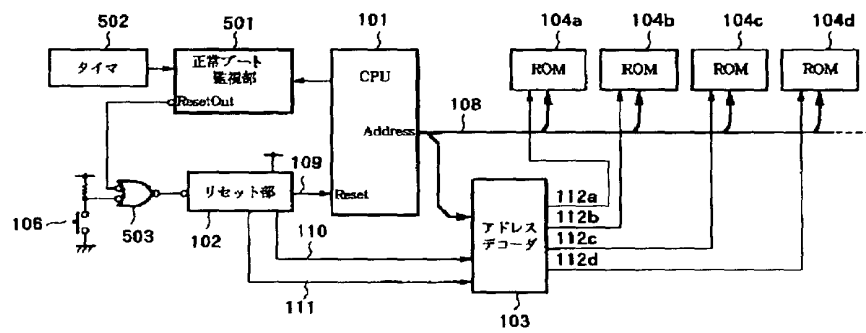
【図 3】



【図 4】



【図 5】



【図 6】

